SREESseminarski

Generated by Doxygen 1.9.4

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 SystemModel Namespace Reference

Classes

- struct AdmittanceMatrix
- class Bus
- class SystemModel

Typedefs

- using fi = std::pair< std::function< double(std::vector< double >)>, std::function< double(std::vector< double >)> >
- using dfidx = std::pair< std::vector< std::function< double(std::vector< double >)> >, std::vector< std::function< double(std::vector< double >)> > >

Enumerations

- enum class TypeOfBus { Slack , PV , PQ }
- enum class ThreePhaseLoadConfigurationsType { Star , GroundedStar , Delta }

Functions

- std::ostream & operator<< (std::ostream &stream, const SystemModel &systemModel)

  Output stream operator overload

4.1.1 Typedef Documentation

4.1.1.1 dfidx

```
using SystemModel::dfidx = typedef std::pair<std::vector<std::function<double(std::vector<double>)>
>, std::vector<std::function<double(std::vector<double>)> >>
```

4.1.1.2 fi

```
using SystemModel::fi = typedef std::pair<std::function<double(std::vector<double>)>, std↩
::function<double(std::vector<double>)> >
```

4.1.2 Enumeration Type Documentation

4.1.2.1 ThreePhaseLoadConfigurationsType

```
enum class SystemModel::ThreePhaseLoadConfigurationsType  [strong]
```

Enumerator

Star	
GroundedStar	
Delta	

4.1.2.2 TypeOfBus

```
enum class SystemModel::TypeOfBus  [strong]
```

Enumerator

Slack	
PV	
PQ	

4.1.3 Function Documentation

4.1.3.1 operator<<()

```
std::ostream & SystemModel::operator<< (
            std::ostream & stream,
            const SystemModel & systemModel )
```

Output stream operator overload

Parameters

stream	Output stream object
systemModel	SystemModel object to be printed to the stream

Returns

Chapter 5

Class Documentation

5.1 SystemModel::AdmittanceMatrix Struct Reference

`#include <systemModel.h>`

Public Attributes

- std::vector< uint8_t > columnIndex
- std::vector< uint8_t > rowIndex
- std::vector< std::complex< double > > values

5.1.1 Member Data Documentation

5.1.1.1 columnIndex

```
std::vector<uint8_t> SystemModel::AdmittanceMatrix::columnIndex
```

5.1.1.2 rowIndex

```
std::vector<uint8_t> SystemModel::AdmittanceMatrix::rowIndex
```

5.1.1.3 values

```
std::vector<std::complex<double> > SystemModel::AdmittanceMatrix::values
```

The documentation for this struct was generated from the following file:

- systemModel.h

5.2 SystemModel::Bus Class Reference

`#include <systemModel.h>`

Public Member Functions

- Bus (TypeOfBus typeOfBus)
- TypeOfBus getTypeOfBus () const
- void setVoltageMagnitude (double voltageMagnitude)

  Sets the value at which the voltage amplitude for the given bus should be maintained.
- void setVoltagePhase (double voltagePhase)

  Sets the value at which the voltage phase for the given bus should be maintained.
- void setActivePower (double activePower)

  Sets the value at which the active power for the given bus should be maintained.
- void setReactivePower (double reactivePower)

  Sets the value at which the rective power for the given bus should be maintained.
- std::optional< double > getVoltageMagnitude () const

  Gets the value at which the voltage magnitude for the given bus should be maintained.
- std::optional< double > getVoltagePhase () const

  Gets the value at which the voltage phase for the given bus should be maintained.
- std::optional< double > getActivePower () const

  Gets the value at which the active power for the given bus should be maintained.
- std::optional< double > getReactivePower () const

  Gets the value at which the rective power for the given bus should be maintained.

5.2.1 Constructor & Destructor Documentation

5.2.1.1 Bus()

```
SystemModel::Bus::Bus (
            TypeOfBus typeOfBus )  [inline]
```

5.2.2 Member Function Documentation

5.2.2.1 getActivePower()

```
std::optional< double > SystemModel::Bus::getActivePower ( ) const
```

Gets the value at which the active power for the given bus should be maintained.

Returns

Value of active power for the bus

5.2.2.2 getReactivePower()

```
std::optional< double > SystemModel::Bus::getReactivePower ( ) const
```

Gets the value at which the rective power for the given bus should be maintained.

Returns

Value of reactive power for the bus

5.2.2.3 getTypeOfBus()

```
TypeOfBus SystemModel::Bus::getTypeOfBus ( ) const  [inline]
```

5.2.2.4 getVoltageMagnitude()

```
std::optional< double > SystemModel::Bus::getVoltageMagnitude ( ) const
```

Gets the value at which the voltage magnitude for the given bus should be maintained.

Returns

Value of voltage magnitude of the bus

5.2.2.5 getVoltagePhase()

```
std::optional< double > SystemModel::Bus::getVoltagePhase ( ) const
```

Gets the value at which the voltage phase for the given bus should be maintained.

Returns

Value of voltage phase of the bus

5.2.2.6 setActivePower()

```
void SystemModel::Bus::setActivePower (
            double activePower )
```

Sets the value at which the active power for the given bus should be maintained.

Parameters

activePower	Value of active power for the bus

5.2.2.7 setReactivePower()

```
void SystemModel::Bus::setReactivePower (
            double reactivePower )
```

Sets the value at which the rective power for the given bus should be maintained.

Parameters

reactivePower	Value of reactive power for the bus

5.2.2.8 setVoltageMagnitude()

```
void SystemModel::Bus::setVoltageMagnitude (
            double voltageMagnitude )
```

Sets the value at which the voltage amplitude for the given bus should be maintained.

Parameters

voltageMagnitude	Value of voltage magnitude of the bus

5.2.2.9 setVoltagePhase()

```
void SystemModel::Bus::setVoltagePhase (
            double voltagePhase )
```

Sets the value at which the voltage phase for the given bus should be maintained.

Parameters

voltagePhase	Value of voltage phase of the bus

The documentation for this class was generated from the following files:

- systemModel.h
- systemModel.cpp

5.3 SystemModel::SystemModel Class Reference

`#include <systemModel.h>`

Public Member Functions

- SystemModel (uint8_t maxNumberOfBuses)
- AdmittanceMatrix getAdmittanceMatrix () const
- uint8_t getNumberOfBuses () const
- Bus & getBus (uint8_t busNumber)

  Gets the bus with the given bus number
- void addBus (TypeOfBus typeOfBus)

  Adds a bus to the system
- void addLoad (uint8_t busNumber, double activePower, double reactivePower)

  Adds a load to a bus
- void addLine (uint8_t busNumber1, uint8_t busNumber2, double r, double x, double b)

  Adds a line between buses
- void addGenerator (uint8_t busNumber, double voltageMagnitude, double activePower)

  Adds a generator to a bus
- void addSlackGenerator (uint8_t busNumber, double voltageMagnitude, double voltagePhase)

  Adds a generator to the slack bus
- bool hasSlackBeenAssigned () const

  Check whether the slack bus has been assigned
- void addTransformer (uint8_t busNumber1, uint8_t busNumber2, double r, double x, double g, double b, double n)

  Adds a transformer between buses
- void addCapacitorBank (uint8_t busNumber, double c, ThreePhaseLoadConfigurationsType configuration↩ Type)

  Adds a capacitor bank to a bus
- fi getBusFunctions (uint8_t busNumber) const

  Gets the bus functions (fi_P and fi_Q) for the desired bus
- dfidx getDerivativesOfBusFunctions (uint8_t busNumber) const

  Gets the derivates of the bus functions (dfi_P/dx and dfi_Q/dx) for the desired bus (two rows of the Jacobian associated with the given bus functions)

Friends

- std::ostream & operator<< (std::ostream &stream, const SystemModel &systemModel)

5.3.1 Constructor & Destructor Documentation

5.3.1.1 SystemModel()

```
SystemModel::SystemModel::SystemModel (
            uint8_t maxNumberOfBuses ) [inline]
```

5.3.2 Member Function Documentation

5.3.2.1 addBus()

```
void SystemModel::SystemModel::addBus (
            TypeOfBus typeOfBus )
```

Adds a bus to the system

Parameters

typeOfBus	Type of the bus (Slack, PV, PQ) to be added to the system

5.3.2.2 addCapacitorBank()

```
void SystemModel::SystemModel::addCapacitorBank (
            uint8_t busNumber,
            double c,
            ThreePhaseLoadConfigurationsType configurationType )
```

Adds a capacitor bank to a bus

Parameters

busNumber	Ordinal number of the desired bus
c	One phase capacitance of the bank
configurationType	Three phase load configuration type (delta, star, grounded star) of the bank

5.3.2.3 addGenerator()

```
void SystemModel::SystemModel::addGenerator (
            uint8_t busNumber,
            double voltageMagnitude,
            double activePower )
```

Adds a generator to a bus

Parameters

busNumber	Ordinal number of the desired bus
voltageMagnitude	Voltage magnitude on which the given bus should be maintained

<param name="activePower"Active power on which the given bus should be maintained>

5.3.2.4 addLine()

```
void SystemModel::SystemModel::addLine (
            uint8_t busNumber1,
            uint8_t busNumber2,
            double r,
            double x,
            double b )
```

Adds a line between buses

Parameters

busNumber1	Ordinal number of the first bus
busNumber2	Ordinal number of the second bus
r	Series resistance of the transmission line PI equivalent
x	Series reactance of the transmission line PI equivalent
b	Shunt susceptance of the transmission line PI equivalent

5.3.2.5 addLoad()

```
void SystemModel::SystemModel::addLoad (
            uint8_t busNumber,
            double activePower,
            double reactivePower )
```

Adds a load to a bus

Parameters

busNumber	Ordinal number of the desired bus
activePower	Active power drawn by the load
reactivePower	Reactive power drawn by the load

5.3.2.6 addSlackGenerator()

```
void SystemModel::SystemModel::addSlackGenerator (
            uint8_t busNumber,
            double voltageMagnitude,
            double voltagePhase )
```

Adds a generator to the slack bus

Parameters

busNumber	Ordinal number of the desired bus
voltageMagnitude	Voltage magnitude on which the given bus should be maintained
voltagePhase	Voltage phase on which the given bus should be maintained

5.3.2.7 addTransformer()

```
void SystemModel::SystemModel::addTransformer (
            uint8_t busNumber1,
            uint8_t busNumber2,
            double r,
            double x,
            double g,
            double b,
            double n )
```

Adds a transformer between buses

Parameters

busNumber1	0rdinal number of the first bus
busNumber2	0rdinal number of the second bus
r	Series resistance of the transformer PI equivalent
x	Series reactance of the transformer PI equivalent
g	Shunt conductance of the transformer PI equivalent
b	Shunt susceptance of the transformer PI equivalent
n	Transformer turns ratio

5.3.2.8 getAdmittanceMatrix()

```
AdmittanceMatrix SystemModel::SystemModel::getAdmittanceMatrix ( ) const  [inline]
```

5.3.2.9 getBus()

```
SystemModel::Bus & SystemModel::SystemModel::getBus (
            uint8_t busNumber )
```

Gets the bus with the given bus number

Parameters

busNumber	Ordinal number of the desired bus

Returns

Bus with the given bus number

5.3.2.10 getBusFunctions()

```
SystemModel::fi SystemModel::SystemModel::getBusFunctions (
            uint8_t busNumber ) const
```

Gets the bus functions (fi_P and fi_Q) for the desired bus

Parameters

busNumber	Ordinal number of the desired bus

Returns

Bus functions for the given bus in the form of std::pair of functions, where both functions have a std::vector of doubles as parameters and return a double

5.3.2.11 getDerivativesOfBusFunctions()

```
SystemModel::dfidx SystemModel::SystemModel::getDerivativesOfBusFunctions (
            uint8_t busNumber ) const
```

Gets the derivates of the bus functions (dfi_P/dx and dfi_Q/dx) for the desired bus (two rows of the Jacobian associated with the given bus functions)

Parameters

busNumber	Ordinal number of the desired bus

Returns

Derivatives of the bus functions for the given bus in the form of std::pair of std::vector-s of functions, where both functions have a std::vector of doubles as parameters and return a double

5.3.2.12 getNumberOfBuses()

```
uint8_t SystemModel::SystemModel::getNumberOfBuses ( ) const  [inline]
```

5.3.2.13 hasSlackBeenAssigned()

```
bool SystemModel::SystemModel::hasSlackBeenAssigned ( ) const
```

Check whether the slack bus has been assigned

Returns

True if the slack bus has been assigned and false otherwise

5.3.3 Friends And Related Function Documentation

5.3.3.1 operator<<

```
std::ostream & operator<< (
            std::ostream & stream,
            const SystemModel & systemModel )  [friend]
```

The documentation for this class was generated from the following files:

- systemModel.h
- systemModel.cpp

Chapter 6

File Documentation

6.1 main.cpp File Reference

```
#include <iostream>
#include "systemModel.h"
```

Functions

- int main ()

6.1.1 Function Documentation

6.1.1.1 main()

```
int main ( )
```

6.2 systemModel.cpp File Reference

```
#include "systemModel.h"
#include <stdexcept>
#include <algorithm>
#include <iomanip>
#include <cmath>
```

Macros

- #define PI 4 * std::atan(1.0)

6.2.1 Macro Definition Documentation

6.2.1.1 PI

```
#define PI 4 * std::atan(1.0)
```

6.3 systemModel.h File Reference

```
#include <vector>
#include <complex>
#include <optional>
#include <functional>
#include <utility>
```

Classes

- struct SystemModel::AdmittanceMatrix
- class SystemModel::Bus
- class SystemModel::SystemModel

Namespaces

- namespace SystemModel

Typedefs

- using SystemModel::fi = std::pair< std::function< double(std::vector< double >)>, std::function< double(std::vector< double >)> >
- using SystemModel::dfidx = std::pair< std::vector< std::function< double(std::vector< double >)> >, std::vector< std::function< double(std::vector< double >)> > >

Enumerations

- enum class SystemModel::TypeOfBus { SystemModel::Slack , SystemModel::PV , SystemModel::PQ }
- enum class SystemModel::ThreePhaseLoadConfigurationsType { SystemModel::Star , SystemModel::GroundedStar , SystemModel::Delta }

Functions

- std::ostream & SystemModel::operator<< (std::ostream &stream, const SystemModel &systemModel)

  Output stream operator overload

6.4 systemModel.h

Go to the documentation of this file.

```
1 #pragma once
2 #include <vector>
3 #include <complex>
4 #include <optional>
5 #include <functional>
6 #include <utility>
7
8
9 namespace SystemModel {
10     using fi = std::pair<std::function<double(std::vector<double>)>,
       std::function<double(std::vector<double>)>;
11
12
13
14     using dfidx = std::pair<std::vector<std::function<double(std::vector<double>)>,
       std::vector<std::function<double(std::vector<double>)>>;
15
16
17
18     enum class TypeOfBus { Slack, PV, PQ };
19
20
21
22     enum class ThreePhaseLoadConfigurationsType { Star, GroundedStar, Delta };
23
24
25
26     struct AdmittanceMatrix {
27         std::vector<uint8_t> columnIndex;
28         std::vector<uint8_t> rowIndex;
29         std::vector<std::complex<double>> values;
30     };
31
32
33
34     class Bus {
35         TypeOfBus typeOfBus;
36         std::optional<double> voltageMagnitude;
37         std::optional<double> voltagePhase;
38         std::optional<double> activePower;
39         std::optional<double> reactivePower;
40     public:
41         Bus(TypeOfBus typeOfBus) : typeOfBus{ typeOfBus } {}
42
43         TypeOfBus getTypeOfBus() const {
44             return typeOfBus;
45         }
46
47         void setVoltageMagnitude(double voltageMagnitude);
48
49         void setVoltagePhase(double voltagePhase);
50
51         void setActivePower(double activePower);
52
53         void setReactivePower(double reactivePower);
54
55         std::optional<double> getVoltageMagnitude() const;
56
57         std::optional<double> getVoltagePhase() const;
58
59         std::optional<double> getActivePower() const;
60
61         std::optional<double> getReactivePower() const;
62     };
63
64
65
66     class SystemModel {
67             AdmittanceMatrix admittanceMatrix;
68             uint8_t numberOfBuses{};
69             std::vector<Bus> buses;
70             const uint8_t maxNumberOfBuses;
71             bool checkForConnectionBetweenToBuses(uint8_t busNumber1, uint8_t busNumber2) const;
72         public:
73             SystemModel(uint8_t maxNumberOfBuses) : maxNumberOfBuses{ maxNumberOfBuses } {}
74
75             AdmittanceMatrix getAdmittanceMatrix() const {
76                 return admittanceMatrix;
77             }
78
79             uint8_t getNumberOfBuses() const {
80                 return numberOfBuses;
```

```
81              }
82
83              Bus& getBus(uint8_t busNumber);
84
85              void addBus(TypeOfBus typeOfBus);
86
87              void addLoad(uint8_t busNumber, double activePower, double reactivePower);
88
89              void addLine(uint8_t busNumber1, uint8_t busNumber2, double r, double x, double b);
90
91              friend std::ostream& operator «(std::ostream& stream, const SystemModel& systemModel);
92
93              void addGenerator(uint8_t busNumber, double voltageMagnitude, double activePower);
94
95              void addSlackGenerator(uint8_t busNumber, double voltageMagnitude, double voltagePhase);
96
97              bool hasSlackBeenAssigned() const;
98
99              void addTransformer(uint8_t busNumber1, uint8_t busNumber2, double r, double x, double g,
    double b, double n);
100
101              void addCapacitorBank(uint8_t busNumber, double c, ThreePhaseLoadConfigurationsType
    configurationType);
102
103              fi getBusFunctions(uint8_t busNumber) const;
104
105              dfidx getDerivativesOfBusFunctions(uint8_t busNumber) const;
106     };
107
108
109
110     std::ostream& operator «(std::ostream& stream, const SystemModel& systemModel);
111 }
```

Index